HLA Baseline Definition

Overview, Rules, and Interface Specification

Dr. Judith Dahmann
Chief Scientist
Defense Modeling & Simulation Office
(703) 998-0660 fax (703) 998-0667
jdahmann@dmso.mil

Topics

- **Overview of HLA**
- **Rules**
- **Runtime Interface Specification**

Overview

High Level Architecture

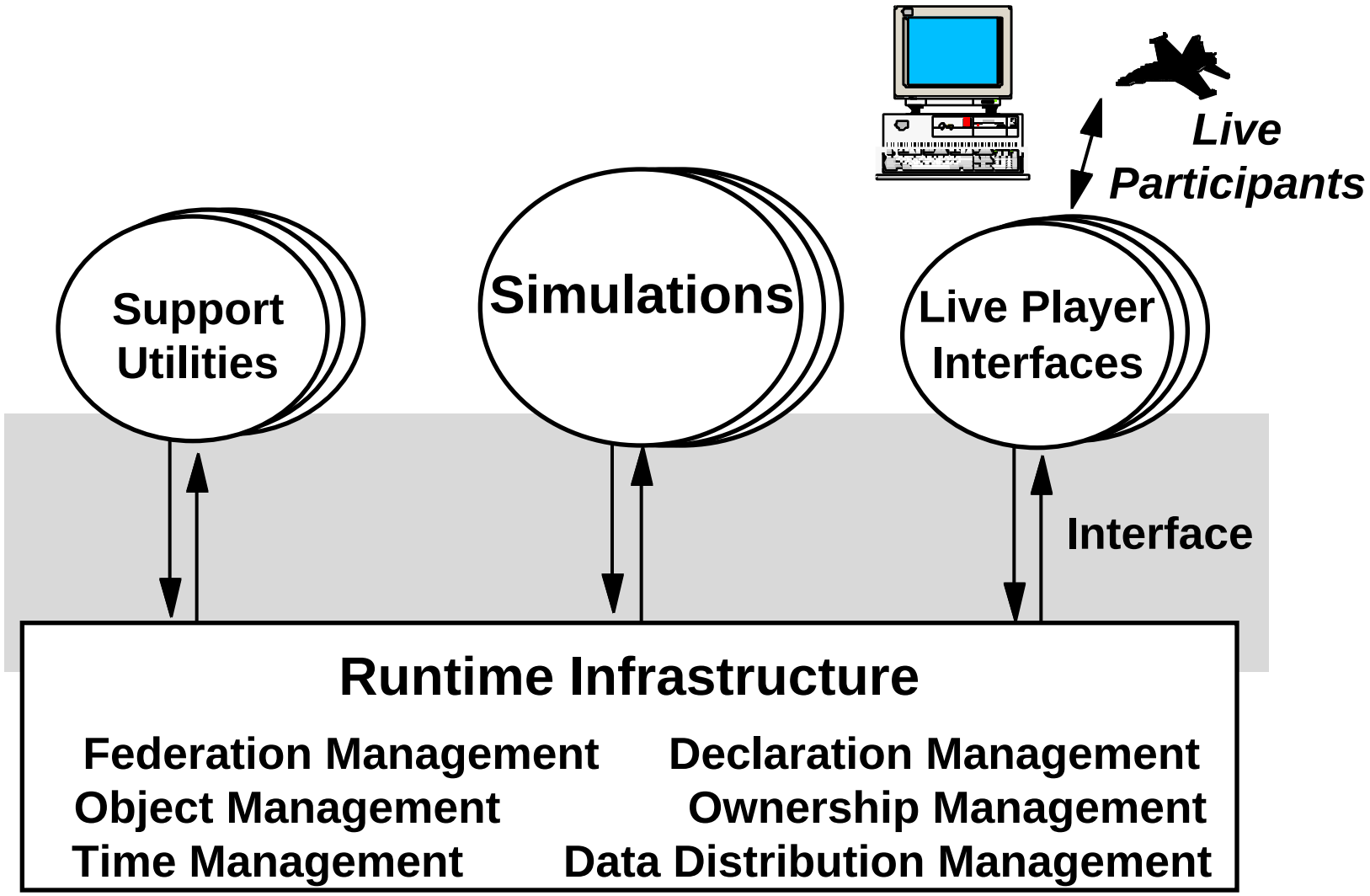
Major functional elements, interfaces, and design rules, pertaining to all DoD simulation applications, and providing a common framework within which specific system architectures can be defined

HLA is the Technical Architecture for DoD Simulations

Defining the HLA

- **HLA Rules**
  - **A set of rules which must be followed to achieve proper interaction of simulations in a federation. These describe the responsibilities of simulations and of the runtime infrastructure in HLA federations.**
- **Interface Specification**
  - **Definition of the interface functions between the runtime infrastructure and the simulations subject to the HLA.**
- **Object Model Template**
  - **The prescribed common method for recording the information contained in the required HLA Object Model for each federation and simulation.**

Functional View of the Architecture



HLA Object Models and OMT

- **Federation Object Model (FOM)**
  - **A description of all shared information (objects, attributes, associations, and interactions) essential to a particular federation**
- **Simulation Object Model (SOM)**
  - **Describes objects, attributes and interactions in a particular simulation which *can* be used externally in a federation**
- **Object Model Template (OMT)**
  - **Provides a common framework for HLA object model documentation**
  - **Fosters interoperability and reuse of simulations and simulation components via the specification of a common representational framework**

Rules

HLA Rules

- **Ten basic rules that define the responsibilities and relationships among the components of an HLA federation**
  - **Five rules apply to federations**
  - **Five rules apply to federates**

Federation Rules

- **Rule 1:**
  - **Federations shall have an HLA Federation Object Model (FOM), documented in accordance with the HLA Object Model Template (OMT).**
- **Rule 2:**
  - **In a federation, all object representation shall be in the federates, not in the runtime infrastructure (RTI).**
- **Rule 3:**
  - **During a federation execution, all exchange of FOM data among federates shall occur via the RTI.**

Federation Rules

- **Rule 4:**
  - **During a federation execution, federates shall interact with the runtime infrastructure (RTI) in accordance with the HLA interface specification.**
- **Rule 5:**
  - **During a federation execution, an attribute of an instance of an object shall be owned by only one federate at any given time.**

Federate Rules

- **Rule 6:**
  - **Federates shall have an HLA Simulation Object Model (SOM), documented in accordance with the HLA Object Model Template (OMT).**
    - **Each simulation must describe the functionality it is able to provide to a federation in OMT terms**
    - **All SOM objects, attributes and interactions may not be used in any given federation**
      - **SOM describes the array of options available**

Federate Rules

- **Rules 7 - 9: Federates have to abide by the provisions of their SOM**
  - **Federates shall be able to update and/or reflect any attributes of objects in their SOM and send and/or receive SOM object interactions externally, as specified in their SOM. (Rule 7)**
  - **Federates shall be able to transfer and/or accept ownership of attributes dynamically during a federation execution, as specified in their SOM. (Rule 8)**
  - **Federates shall be able to vary the conditions (e.g., thresholds) under which they provide updates of attributes of objects, as specified in their SOM. (Rule 9)**

Federate Rules

- **Rule 10: Time Management**
  - **Federates shall be able to manage local time in a way which will allow them to coordinate data exchange with other members of a federation.**
    - **Simulations in a federation must manage time so that there appears to be one clock**
    - **Internally, a simulation manages time any way it wishes, as long is it meets commitments to other simulations in the federation**

Interface Specification

Interface Specification

- **Provides a specification of the functional interfaces between federates and the RTI**
  - **63 interfaces in six service groups**
- **Includes:**
  - **Name and Descriptive Text**
  - **Supplied Parameters**
  - **Returned Parameters**
  - **Pre-conditions**
  - **Post-conditions**
  - **Exceptions**
  - **Related Services**

  and Application Programmers Interface in CORBA IDL
- **Language specific APIs are planned as annexes**
  - **C++ is currently available (will be implemented with RTI 1.0)**

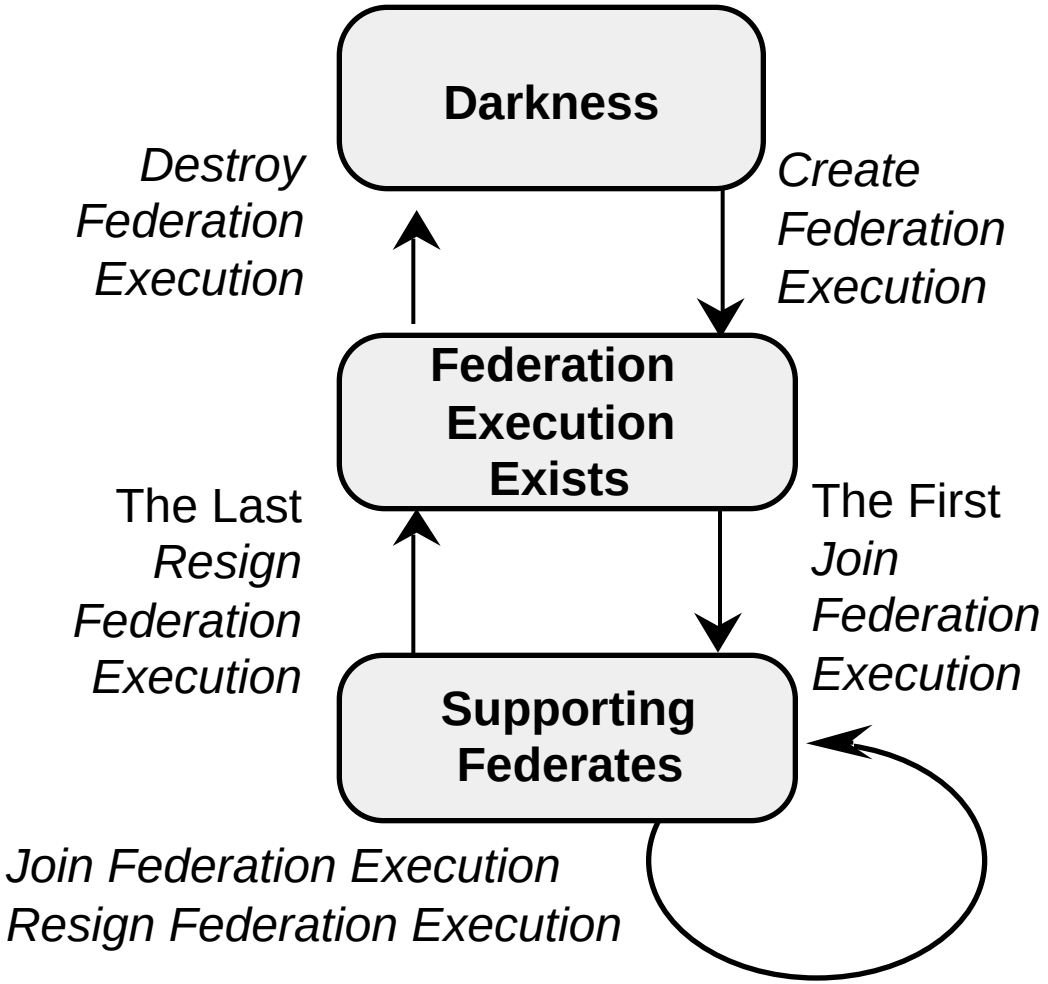
Six HLA Runtime Infrastructure Service Groups

- Federation Management
- Declaration Management
- Object Management
- Ownership Management
- Time Management
- Data Distribution Management

Federation Management

- **Coordinate federation-wide activities throughout the life of a federation execution**
  - **Used by federates to manage a federation execution to meet their needs**
  - **Includes RTI initialization data**
    - **Initializing name space, transportation and ordering defaults and routing space names and dimensions**
- **Interface functions include**
  - **Creation and destruction of a federation execution**
  - **Joining and resigning of a federate**
  - **Coordination of federation saves**
  - **Pausing and resuming a federation execution**

Federation Management



Declaration Management

- **Allow federates to specify the types of data they will send or receive by object class and attribute name and by interaction class from the FOM**
- **Interface functions include specification of:**
  - **Data to be sent:**
    - **Object classes and attributes and interaction classes that the federate is able to update or send**
  - **Data to be received:**
    - **Object classes and attributes and interaction classes that the federate is interested to receive**
  - **Controls on data to be sent:**
    - **Feedback to the federates from the RTI when attribute updates and interactions should be sent given the interest in those by other federates**

Object Management

- **Supports creation, modification, and deletion of objects, their attributes and the interactions they produce**
- **Interface functions include**
  - **Federate requests for IDs**
  - **Registering and discovering objects**
  - **Updating and reflecting object attributes**
  - **Sending and receiving interactions**
  - **Deleting and removing objects**
  - **Changing default transportation and event ordering types**

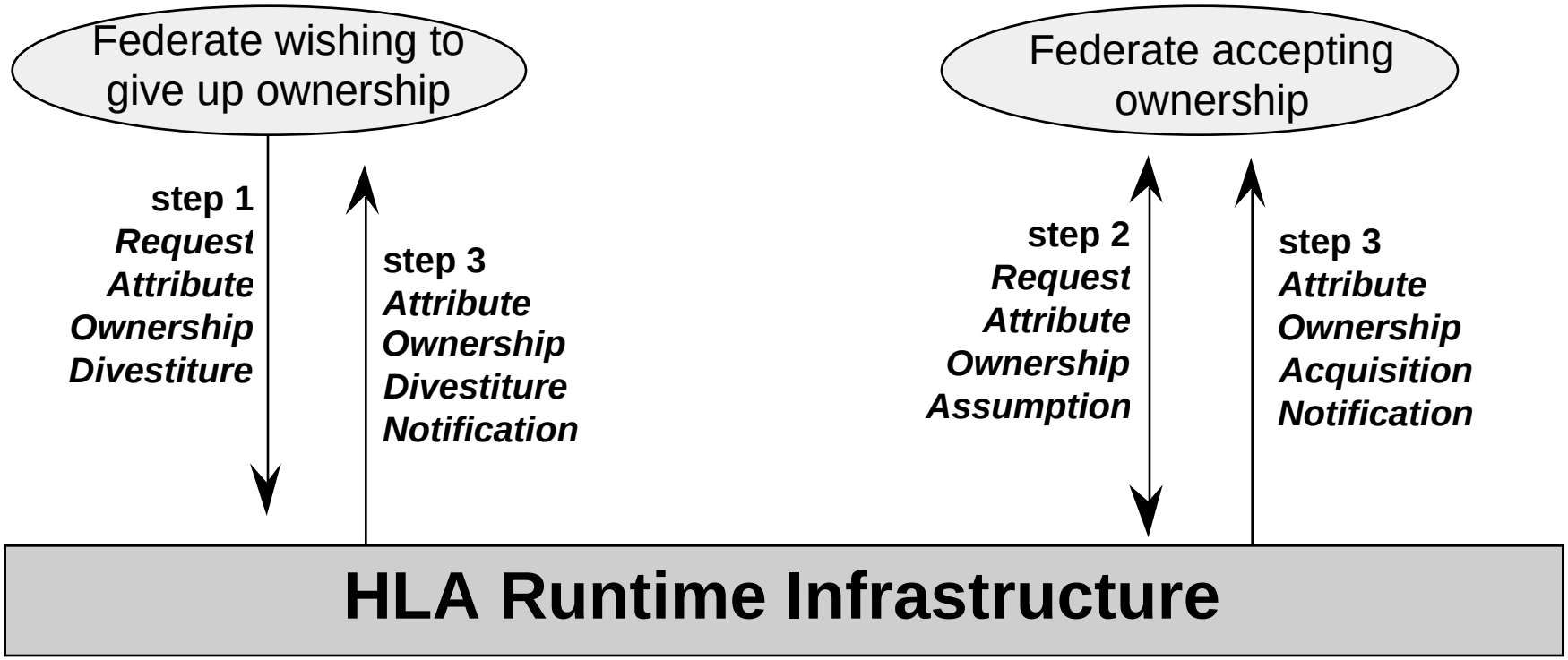
Time Management

- **Control advancement of federates along with federation time**
  - **Coordinated with object management services to support causal behavior across the federation**
  - **Designed to support federates with different ordering and delivery requirements**
- **Interface functions include**
  - **Request current values of time**
    - **federation time, federate's logical time (LT), lower bound time stamp (LBTS), minimum next event time**
  - **Set and request lookahead**
  - **Time advance request, next event and flush queue request, and grant**

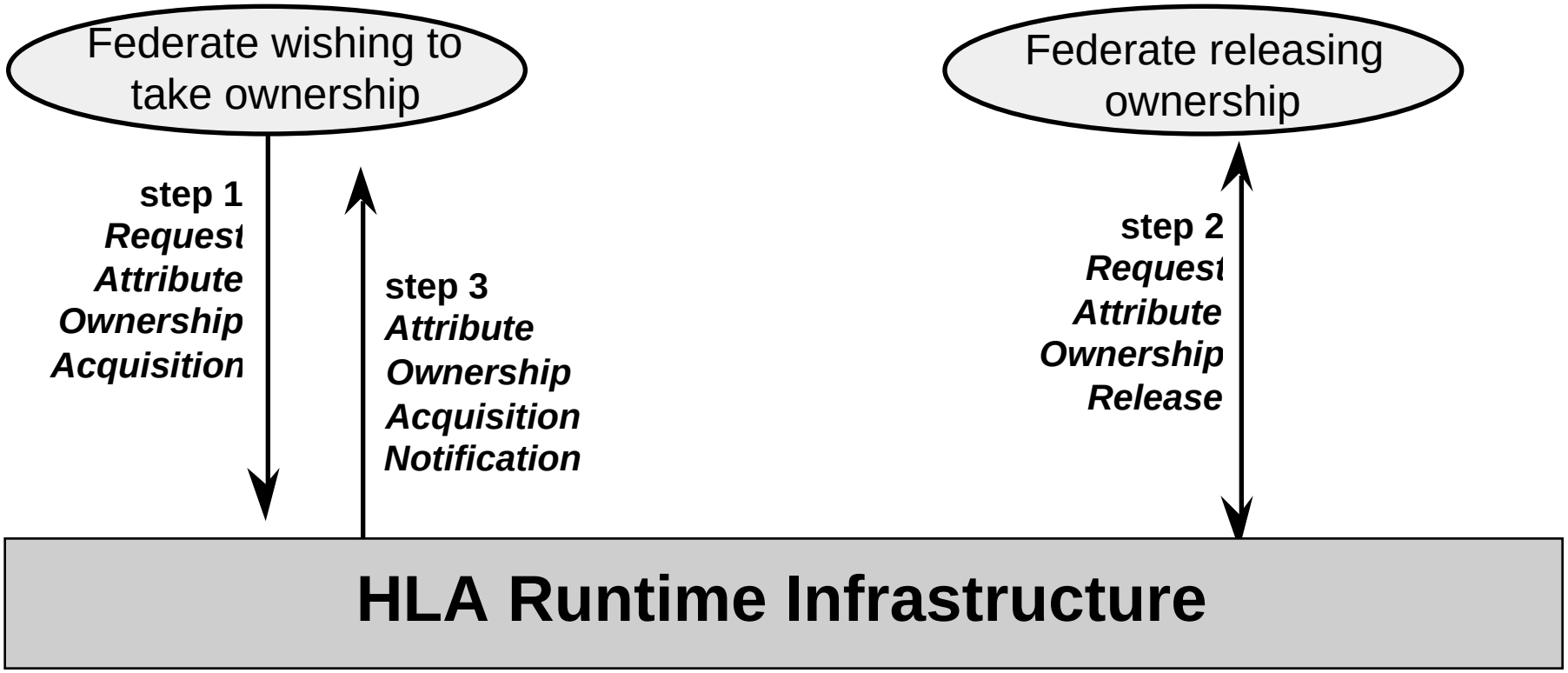
Ownership Management

- **Allow federates to transfer ownership of object attributes**
  - **Federates transfer ownership based on federation execution design plans and the RTI arbitrates transactions**
  - **Offers both ‘push’ or ‘pull’ based transactions**
    - **Acquisition requires current publication and subscription declarations for attribute**
- **Interface functions include**
  - **Request ownership divestiture and assumption**
  - **Request ownership acquisition and release**
  - **Notification of divestiture and acquisition**
  - **Query attribute ownership**

Divesting Ownership



Requesting Ownership



Data Distribution Management

- **Allow federates to specify the distribution conditions for the specific data they send or expect to receive**
  - **RTI uses this information to route data as specified in declaration management services**
  - **Not bound by FOM, data distribution can be managed based on other characteristics of objects important to particular federation execution**
  - **Federation design creates 'routing spaces' for use during runtime; these are specified at federation creation time**
- **Interface functions include**
  - **Create and modify 'update' and 'subscription' regions to bound routing space**
  - **Associate update regions with specific object instances**
  - **Change thresholds for changing regions**

The Role of the Federate in DDM

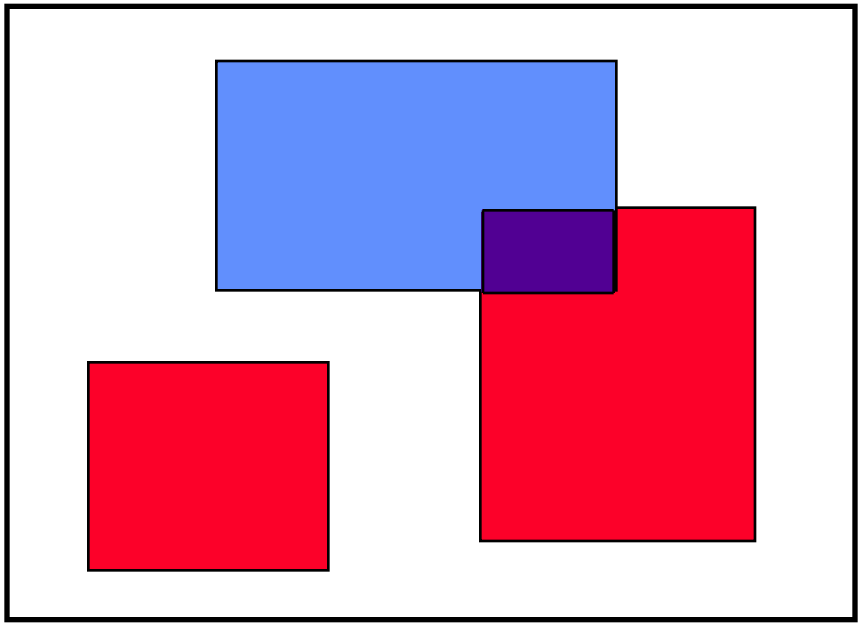
- **`Create Subscription Region`**
  - **Specify conditions under which they expect to receive the object state data and interactions they specified using declaration management services (`Subscribe Object Class Attribute` and `Subscribe Interaction Class`) and**
- **`Create Update Region`**
- **`Associate Update Region` (with an object instance or interaction)**
  - **Specify conditions under which they are providing data (characteristics of object or interaction which map to dimension of routing space fall with region bounds)**
- **`Modify Region or Associate Update Region`**
  - **As the state of the objects change, the federate may need to either adjust the bounds on the associated regions or change the association to another region**

The Role of the RTI in DDM

- **The routing space, regions, and association data is used by the RTI to distribute data**
- **When an update region and subscription regions of different federates overlap**
  - **the RTI ensures that the attribute updates and interactions associated with that update region are routed to federates with subscription regions which overlap the sender's update region**
- **`Change Thresholds`**
  - **The RTI provides feedback to federate on the amount of change in extents which will lead to data distribution changes**

Illustration of DDM Services

Two Dimensional Interest Space

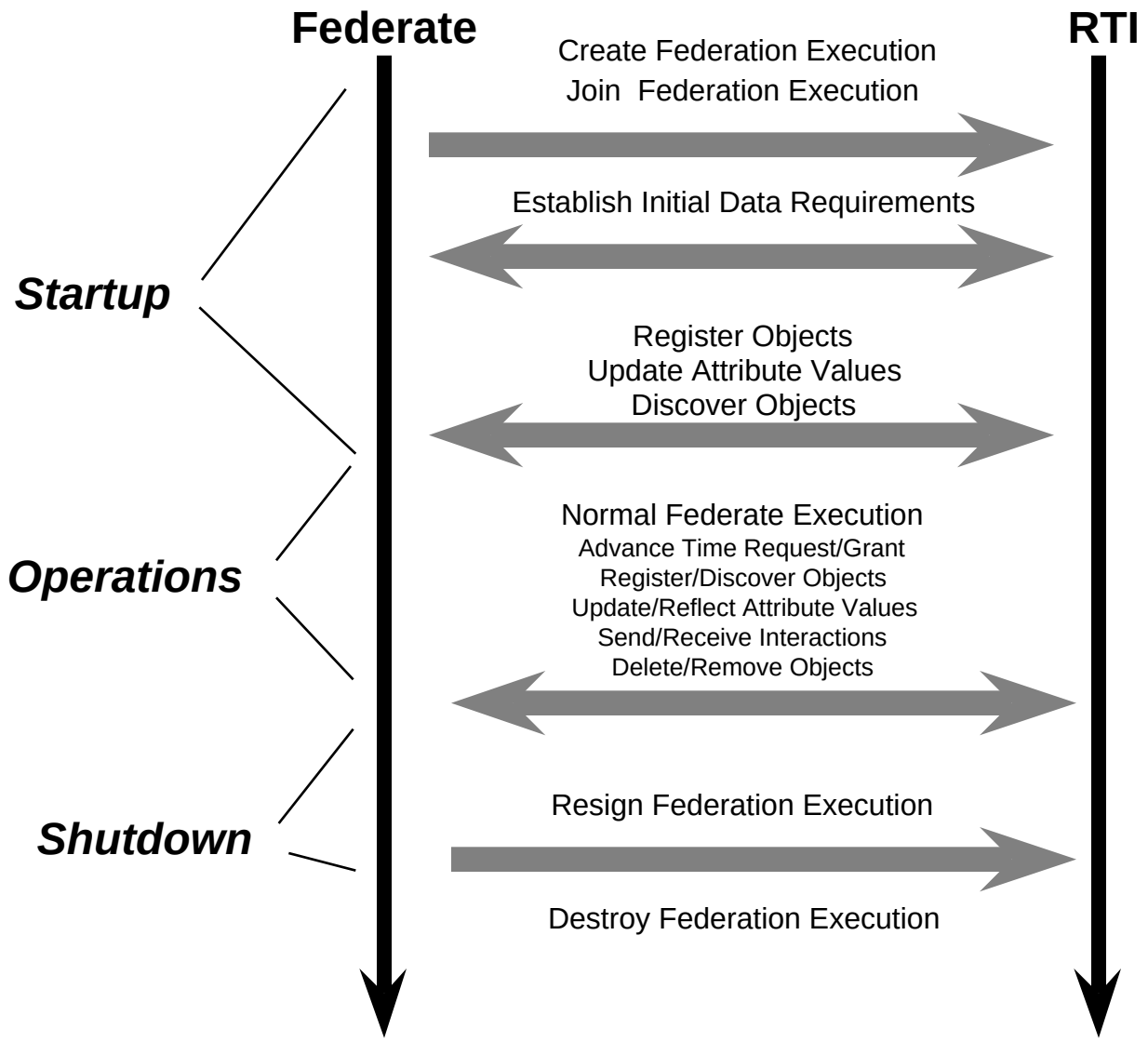


Update Region

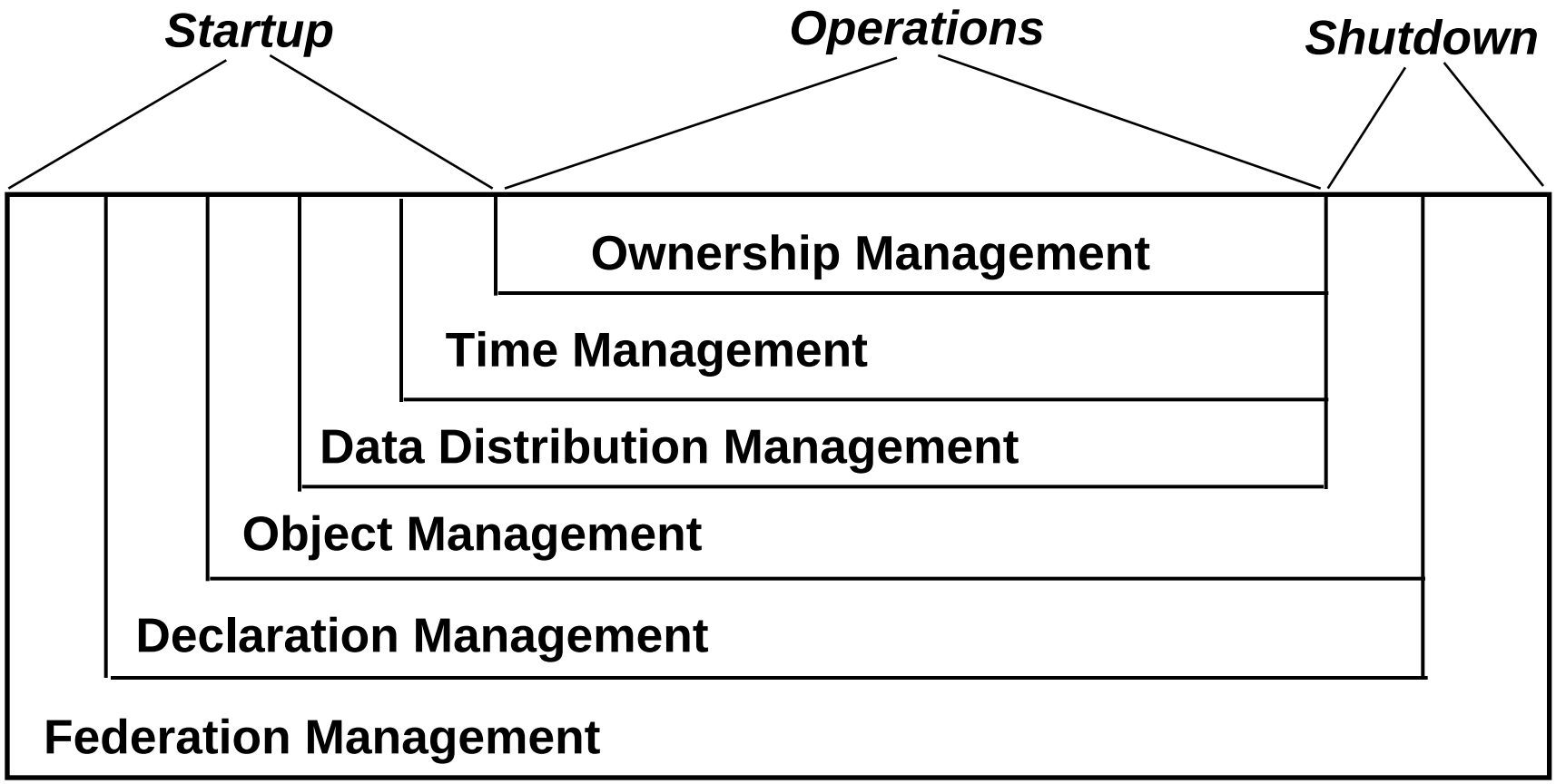
Subscription Region

Overlap Region - Published Data Sent to Subscribing Federate

Overview of Federation Execution Life Cycle



Use of Interface Over Life Cycle of a Federation Execution



Startup: Federation Management

Manage a federation execution throughout the life cycle

- **`Create Federation Execution`**
  - **Initialize the RTI with Federation Specific Information**
    - **Establish class, attribute, interaction names and hierarchies, as defined in FOM**
    - **Set default vales for ordering and transport services**
    - **Establish names and dimensions of routing spaces (if using data distribution management services)**
- **`Join Federation Execution`**
  - **Affiliates a federate with the federation execution**

Startup: Declaration Management

Federate desires to generate and receive data

- **`Publish Object Class` and `Publish Interaction Class`**
  - **Informs RTI of ability to update attributes of classes of objects or send classes of interactions**
- **`Subscribe Object Class Attribute` and `Subscribe Interaction Class`**
  - **Informs RTI of desire to discover object attributes and classes of interactions**

Startup: Object Management

Supports creation, modification and deletion of objects

- **`Request ID`**
  - **Federate requests unique ID numbers from RTI**
- **`Register Object`**
  - **Links an object ID with an instance of an object**
- **`Update Attribute Values`**
  - **Provides current attribute values of an instance of an object**
- **`Discover Object`**
- **`Reflect Attribute Values`**
  - **RTI informs federates with a declared interest in attributes objects of their existence and their current values**

Startup: Data Distribution Management

Supports management of data distribution

- **Create Update Region**
- **Create Subscription Region**
- **Associate Update Region**
  - **Federate identifies subsets of routing spaces which meet data distribution needs of the objects it represents**

Startup: Time Management

Controls federate advance along with federation time

- **Set default transportation event ordering type for object classes and interactions in RID**
- **`Set Lookahead`**
  - **If using event ordering services for any objects**

Operations: Object and Time Management

- **Object Management**
  - **Update Attributes; Send Interactions**
    - Federates export event data
  - **Reflect Attributes and Receive Interactions**
    - RTI delivers data to federates
- **Time Management**
  - **Time Advance Request; Next Event Request, Flush Queue Request**
    - Federates request time ordered event data from RTI
  - **Time Advance Grant**
    - RTI delivers events up to time requested along permission to move forward
  - **Request .... Time**
    - Federate queries RTI for time value

Operations: Declaration and Data Distribution Mgt

- **Declaration Management**
  - **Publish Object Class or Interaction Class**
  - **Subscribe Object Class Attribute or Interaction Class**
    - **Federate can reissue data declarations, overriding previous**
  - **Control Updates and Interactions**
    - **RTI can inform federates of changes in need to update or send**
- **Data Distribution Management**
  - **Modify Region; Change Thresholds**
  - **Associate Update Region**
  - **Create Update or Subscription Region**
    - **Federate can change data routing by changing bound on regions, associating an object with a new region or creating or deleting regions**

Operations: Federation and Ownership Management

- **Federation Management**
  - **Request Pause, Initiate Pause, Pause Achieved**
  - **Request Resume, Initiate Resume, Resume Achieved**
    - **Federates can request that the federation pause operations and subsequently resume; RTI coordinates with all federates**
  - **Request Federation Save, Initiate Federation Save, Federation Save Achieved**
  - **Request Restore, Initiate Restore, Restore Achieved**
    - **Federates can request that the federation save state and subsequently restore to a particular saved state; RTI coordinates with all federates**
- **Ownership Management**
  - **Federates can transfer ownership**

Shutdown

- **Federation Management**
  - **`Resign Federation Execution`**
    - **Indicates that a Federate chooses to cease participation**
    - **Triggers ownership transfer for attributes owned by resigning federate or deletes these attributes (options on the service call)**
  - **`Destroy Federation Execution`**
    - **Removes this federation execution from all RTI support; assumes all federates have resigned**